# Socially-aware Multiagent Learning towards Socially Optimal Outcomes

**Xiaohong Li** [1] and **Chengwei Zhang** [1] and **Jianye Hao** [1] and **Karl Tuyls** [2] and **Siqi Chen** [3]

**Abstract.** In multiagent environments, the capability of learning is important for an agent to behave appropriately in face of unknown opponents and dynamic environment. From the system designer's perspective, it is desirable if the agents can learn to coordinate towards socially optimal outcomes, while also avoiding being exploited by selfish opponents. To this end, we propose a novel gradient ascent based algorithm (SA-IGA) which augments the basic gradient-ascent algorithm by incorporating social awareness into the policy update process. We theoretically analyze the learning dynamics of SA-IGA using dynamical system theory and SA-IGA is shown to have linear dynamics for a wide range of games including symmetric games. The learning dynamics of two representative games (the prisoner's dilemma game and coordination game) are analyzed in details. Based on the idea of SA-IGA, we further propose a practical multiagent learning algorithm, called SA-PGA, based on Q-learning update rule. Simulation results show that SA-PGA agent can achieve higher social welfare than previous social-optimality oriented Conditional Joint Action Learner (CJAL) and also is robust against individually rational opponents by reaching Nash equilibrium solutions.

## 1 Introduction

In multiagent systems, the ability of learning is important for an agent to adaptively adjust its behaviors in response to coexisting agents and unknown environments in order to optimize its performance. Multiagent learning algorithms have received extensive investigation in the literature, and lots of learning strategies [5, 13, 3] have been proposed to facilitate coordination among agents.

The multi-agent learning criteria proposed in [4] require that an agent should be able to converge to a stationary policy against some class of opponents (*convergence*) and the best-response policy against any stationary opponent (*rationality*). If both agents adopt a rational learning strategy in the context of repeated games and also their strategies converge, then they will converge to a Nash equilibrium of the stage game. Indeed, convergence to Nash equilibrium has been the most commonly accepted goal to pursue in multiagent learning literature. Until now, a number of gradient-ascent based multiagent learning algorithms [17, 4, 1, 21] have been sequentially proposed towards converging to Nash equilibrium with improved convergence performance and more relaxed assumptions (less information is required). Under the same direction, another well-studied family of multiagent learning strategies is based on reinforcement learning (e.g., Q-learning [20]). Representative examples include distributed Q-learning in cooperative games [10], minimax Q-learning in zero-sum games [11], Nash Q-learning in general-sum games [9], and other extensions [12, 5], to name just a few.

| 1's payoff<br>2's payoff | | Agent 2's actions | |
|---|---|---|---|
| | | C | D |
| Agent 1's actions | C | 3/3 | 0/5 |
| | D | 5/0 | 1/1 |

**Table 1**: The Prisoner's Dilemma Game

All the aforementioned learning strategies pursue converging to Nash equilibrium under self-play, however, Nash equilibrium solution may be undesirable in many scenarios. One well-known example is the prisoner's dilemma (PD) game shown in Table 1. By converging to the Nash equilibrium $(D, D)$, both agents obtain the payoff of 1, while they could have obtained a much higher payoff of 3 by coordinating on the non-equilibrium outcome $(C, C)$. In situations like the PD game, converging to the socially optimal outcome under self-play would be more preferred. To address this issue, one natural modification for a gradient-ascent learner is to update its policy along the direction of maximizing the sum of all agents' expected payoff instead of its own. However, in an open environment, the agents are usually designed by different parties and may have not the incentive to follow the strategy we design. The above way of updating strategy would be easily exploited and taken advantage by (equilibrium-driven) self-interested agents. Thus it would be highly desirable if an agent can converge to socially optimal outcomes under self-play and Nash equilibrium against self-interested agents to avoid being exploited.

In this paper, we first propose a new gradient-ascent based algorithm (SA-IGA) which augments the basic gradient ascent algorithm by incorporating social awareness into the policy update process. A SA-IGA agent holds a social attitude to reflect its socially-aware degree, which can be adjusted adaptively based on the relative performance between its own and its opponent. A SA-IGA agent seeks to update its policy in the direction of increasing its overall payoff which is defined as the average of its individual and the social payoff weighted by its socially-aware degree. We theoretically show that for a wide range of games (e.g., symmetric games), the dynamics of SA-IGAs under self-play exhibits linear characteristics. For general-sum games, it may exhibit non-linear dynamics which can still be analyzed numerically. The learning dynamics of two representative games (PD game and coordination game) are analyzed in details. Like previous theoretical multiagent learning algorithms, SA-IGA

[1] Tianjin University, China. Email:{xiaohongli, chenvy}@tju.edu.cn, jianye.hao@tju.edu.cn
[2] University of Liverpool, UK. Email: ktuyls@liverpool.ac.uk
[3] Southwest University, China. Email: siqichen@swu.edu.cn

also requires additional assumption of knowing the opponent's policy and the game structure.

To relax the above assumption, we then propose a practical gradient ascent based multiagent learning strategy, called Socially-aware Policy Gradient Ascent (SA-PGA). SA-PGA relaxes the above assumptions by estimating the performance of its own and the opponent using Q-learning techniques. We empirically evaluate its performance in different types of benchmark games and simulation results show that SA-PGA agent outperforms previous learning strategies in terms of maximizing the social welfare and Nash product of the agents. Besides, SA-PGA is also shown to be robust against individually rational opponents and converges to Nash equilibrium solutions.

The remainder of the paper is organized as follows. Section 2 reviews normal-form game and the basic gradient ascent approach. Section 3 introduces the SA-IGA algorithm and analyzes its learning dynamics theoretically. Section 4 presents the practical multiagent learning algorithm SA-PGA in details. In Section 5, we extensively evaluate the performance of SA-PGA under various benchmark games. Lastly we conclude the paper and point out future directions in Section 6.

## 2 Background

### 2.1 Normal-form games

In a two-player, two-action, general-sum normal-form game, the payoffs for each player $i \in \{r, c\}$ can be specified by a matrix as follows,

$$R_i = \begin{bmatrix} r_{11}^i & r_{12}^i \\ r_{21}^i & r_{22}^i \end{bmatrix}$$

Each player $i$ simultaneously selects an action from its action set $A_i = \{1, 2\}$, and the payoff of each player is determined by their joint actions. For example, if player $r$ selects the pure strategy of action 1 while player $c$ selects the pure strategy of action 2, then player $r$ receives a payoff of $r_{12}^r$ and player $c$ receives the payoff of $r_{12}^c$.

Apart from pure strategies, each player can also employ a mixed strategy to make decisions. A mixed strategy can be represented as a probability distribution over the action set and a pure strategy is a special case of mixed strategies. Let $p_r \in [0, 1]$ and $p_c \in [0, 1]$ denote the probability of choosing action 1 by player $c$ and player $r$ respectively. Given a joint mixed strategy $(p_r, p_c)$, the expected payoffs of player $c$ and player $r$ can be specified as follows,

$$\begin{aligned} V_r(p_r, p_c) =& r_{11}^r p_r p_c + r_{12} p_r (1 - p_c) + r_{21}^r (1 - p_r) p_c \\ &+ r_{22}^r (1 - p_r)(1 - p_c) \\ V_c(p_r, p_c) =& r_{11}^c p_r p_c + r_{12}^c p_r (1 - p_c) + r_{21}^c (1 - p_r) p_c \\ &+ r_{22}^c (1 - p_r)(1 - p_c) \end{aligned} \tag{1}$$

respectively.

A joint strategy is called a Nash Equilibrium (NE), if no player can get a better expected payoff by changing its current strategy unilaterally. Formally, $(p_r^*, p_c^*) \in [0, 1]^2$ is a NE, iff $V_r(p_r^*, p_c^*) \geq V_r(p_r, p_c^*)$ and $V_c(p_r^*, p_c^*) \geq V_c(p_r^*, p_c)$ for any $(p_r, p_c) \in [0, 1]^2$.

### 2.2 Gradient Ascent (GA)

When a game is repeatedly played, an individually rational player updates its strategy towards maximizing its expected payoffs. A player $i$ employing GA-based algorithms updates its policy towards the direction of its expected reward gradient, which can be shown in the following equations.

$$\Delta p_i^{(t+1)} \leftarrow \eta \frac{\partial V_i\left(p^{(t)}\right)}{\partial p_i} \tag{2}$$

$$p_i^{(t+1)} \leftarrow \Pi_{[0,1]}\left(p_i^{(t)} + \Delta p_i^{(t+1)}\right) \tag{3}$$

where parameter $\eta$ is the gradient step size, and $\Pi_{[0,1]}$ is the projection function mapping the input value to the valid probability range of $[0, 1]$, used to prevent the gradient moving the strategy out of the valid probability space. Formally, we have,

$$\Pi_{[0,1]}(x) = argmin_{z \in [0,1]} |x - z| \tag{4}$$

To simplify the notations, let us denote $u_i = r_{11}^i + r_{22}^i - r_{12}^i - r_{21}^i$, $c_i = r_{12}^i - r_{22}^i$ and $d_i = r_{21}^i - r_{22}^i$. For the two-player case, the above way of GA-based updating in Equation 2 and 3 can be represented as follows,

$$p_r^{(t+1)} \leftarrow \Pi_{[0,1]}\left(p_r^{(t)} + \eta\left(u_r p_c^{(t)} + c_r\right)\right) \tag{5}$$

$$p_c^{(t+1)} \leftarrow \Pi_{[0,1]}\left(p_c^{(t)} + \eta\left(u_c p_r^{(t)} + d_c\right)\right) \tag{6}$$

In the case of infinitesimal gradient step size ($\eta \to 0$), the learning dynamics of the players can be modeled as a system of differential equations and analyzed using dynamic system theory [17]. It is proved that the agents will converge to a Nash equilibrium, or if the strategies themselves do not converge, then their average payoffs will nevertheless converge to the average payoffs of a Nash equilibrium [17].

Following [17], various GA-based algorithms have been proposed to improve the convergence performance towards Nash equilibria and representative examples include IGA-WoLF (Win or Learn Fast) [4], Weighted Policy Learner (PWL) [1] and Gradient Ascent With Policy Prediction (IGA-PP) [21]. In contrast, in this work, we seek to incorporate the social awareness into GA-based strategy update and aim at improving the social welfare of the players under self-play rather than pursuing Nash equilibrium solutions. Meanwhile, individually rational behavior is employed when playing against a selfish agent. Similar idea of adaptively behaving differently against different opponents was also employed in previous algorithms [12, 8, 14, 6]. However, all the existing works focus on maximizing an agent's individual payoff against different opponents in different types of games, but do not directly take into consideration the goal of maximizing social welfare (e.g., cooperate in the prisoner's dilemma game).

## 3 Socially-aware Infinitesimal Gradient Ascent (SA-IGA)

In our daily life, people usually do not always behave as a purely individually rational entity and seeks to achieve Nash equilibrium solutions. For example, when two person subjects play a PD game, reaching mutual cooperation may be observed frequently. Similar phenomenon have also been observed in extensive human-subject based experiments in games such as the Public Good game and Ultimatum game, in which human subjects are usually found to obtain much higher payoff by mutual cooperation rather than pursuing Nash equilibrium solutions. If the above phenomenon is transformed into computational models, it indicates that an agent may not only update its policy in the direction of maximizing its own payoff, but also take into consideration other's payoff. We call this type of agents as socially-aware agents.

In this paper, we incorporate the social awareness into the gradient-ascent based learning algorithm. In this way, apart from

learning to maximizing its individual payoff, an agent is also equipped with the social awareness such that it can (1) reach mutually cooperative solutions faced with another socially-aware opponent (self-play); (2) behave in a purely individually rational manner against a purely rational opponent.

Specifically, for each agent $i \in \{r, c\}$,we distinguish two types of expected payoffs, namely $V_i^{idv}$ and $V_i^{soc}$. The payoff $V_i^{idv}(p_r, p_c)$ and $V_i^{soc}(p_r, p_c)$ represent the individual and social payoff (the average payoff of both players) that agent $i$ perceives under the joint strategy $(p_r, p_c)$ respectively. The payoff $V_i^{idv}(p_r, p_c)$ follows the same definition as Equation (1) and the payoff $V_i^{soc}(p_r, p_c)$ can be defined as follows,

$$V_i^{soc}(p_r, p_c) = \frac{1}{2}[V_r^{idv}(p_r, p_c) + V_c^{idv}(p_r, p_c)], \forall i \in \{r, c\} \quad (7)$$

Each agent $i$ adopts a social attitude $w_i$ to reflect its socially-aware degree. The social attitude intuitively models an agent's socially friendly degree towards its partner. Specifically, it is used as the weighting factor to adjust the relative importance between $V_i^{idv}$ and $V_i^{soc}$, and agent $i$'s overall expected payoff is defined as follows,

$$V_i(p_r, p_c) = (1 - w_i) V_i^{idv}(p_r, p_c) + w_i V^{soc}(p_r, p_c), \forall i \in \{r, c\} \quad (8)$$

Each agent $i$ updates its strategy in the direction of maximizing the value of $V_i$. Formally we have,

$$\begin{aligned} \Delta p_i &\leftarrow \eta_p \frac{\partial V_i(p_r, p_c)}{\partial p_i} \\ p_i &\leftarrow \Pi_{[0,1]}(p_i + \Delta p_i) \end{aligned} \quad (9)$$

where parameter $\eta_p$ is the gradient step size of $p_i$. If $w_i = 0$, it means that the agent seeks to maximize its individual payoff only, which is reduced to the case of traditional gradient-ascent updating; if $w = 1$, it means that the agent seeks to maximize the sum of the payoffs of both players.

Finally, each agent $i$'s socially-aware degree is adaptively adjusted in response to the relative value of $V_i^{idv}$ and $V_i^{soc}$ as follows. During each round, if player $i'$s own expected payoff $V_i^{idv}$ exceeds the value of $V^{soc}$, then player $i$ increases its social attitude $w_i$, (i.e., it becomes more social-friendly because it perceives itself to be earning more than the average). Conversely, if $V_i^{idv}$ is less than $V^{soc}$ t, then the agent tends to care more about its own interest by decreasing the value of $w_i$. Formally we have,

$$w_i = \begin{cases} \Pi_{[0,1]}(w_i + \Delta w_i) & \text{if } V_i^{idv} > V_i^{soc} \\ \Pi_{[0,1]}(w_i - \Delta w_i) & \text{if } V_i^{idv} < V_i^{soc} \end{cases} \quad (10)$$

where $\Delta w_i$ is the adjustment step size of $w_i$.

## 3.1 Theoretical Modeling and Analysis of SA-IGA

An important aspect of understanding the behavior of a multiagent learning algorithm is theoretically modeling and analyzing its underlying dynamics [19, 15, 3]. In this section, we first show that the learning dynamics of SA-IGA under self-play can be modeled as a system of differential equations.

Based on the adjustment rules in Eq (9) and (10), the learning dynamics of a SA-IGA agent can be modeled as a set of equations in (11). For ease of exposition, we concentrate on a unconstrained update equations by removing the policy projection function which does not affect our qualitative analytical results. Any trajectory with linear (non-linear) characteristic without constraints is still linear (non-linear) when a boundary is enforced.

$$\begin{aligned} \Delta p_i^{(t+1)} &\leftarrow \eta_p \frac{\partial V_i\left(p_r^{(t)}, p_c^{(t)}\right)}{\partial p_i} \\ \Delta w_i^{t+1} &\leftarrow \eta_w (V_i^{idv} - V^{soc}) \\ p_i^{(t+1)} &\leftarrow p_i^{(t)} + \Delta p_i^{(t+1)} \\ w_i^{(t+1)} &\leftarrow w_i^{(t)} + \Delta w_i^{(t+1)} \end{aligned} \quad (11)$$

Substituting $V_i^{idv}$ and $V_i^{soc}$ by their definitions (Eq. (1) and (7)), the learning dynamics of two SA-IGA agents can be expressed as follows,

$$\begin{aligned} \Delta p_r^{t+1} &= \eta_p \left[\left(u_r + \frac{u_c - u_r}{2} w_r^t\right) p_c^t + \frac{c_c - c_r}{2} w_r^t + c_r\right] \\ \Delta p_c^{t+1} &= \eta_p \left[\left(u_c + \frac{u_r - u_c}{2} w_c^t\right) p_r^t + \frac{d_r - d_c}{2} w_c^t + d_c\right] \\ \Delta w_r^{t+1} &= \\ &\eta_w \left[(u_r - u_c) p_r^t p_c^t + (c_r - c_c) p_r^t + (d_c - d_r) p_c^t + e\right] \\ \Delta w_c^{t+1} &= \\ &-\eta_w \left[(u_r - u_c) p_r^t p_c^t + (c_r - c_c) p_r^t + (d_c - d_r) p_c^t + e\right] \end{aligned} \quad (12)$$

where $u_i = r_{11}^i + r_{22}^i - r_{12}^i - r_{21}^i$, $c_i = r_{12}^i - r_{22}^i$,$d_i = r_{21}^i - r_{22}^i$, and $e = r_{22}^r - r_{22}^c$ with $i \in \{r, c\}$.

As $\eta_p \to 0$ and $\eta_w \to 0$, it is straightforward to show that the above equations become differential. Thus the unconstrained dynamics of the strategy pair and social attitudes as a function of time is modeled by the following system of differential equations:

$$\begin{aligned} \dot{p}_r &= \left(u_r + \frac{u_c - u_r}{2} w_r\right) p_c + \frac{c_c - c_r}{2} w_r + c_r \\ \dot{p}_c &= \left(u_c + \frac{u_r - u_c}{2} w_c\right) p_r + \frac{d_r - d_c}{2} w_c + d_c \\ \dot{w}_r &= \varepsilon \left[(u_r - u_c) p_r p_c + (c_r - c_c) p_r + (d_c - d_r) p_c + e\right] \\ \dot{w}_c &= -\varepsilon \left[(u_r - u_c) p_r p_c + (c_r - c_c) p_r + (d_c - d_r) p_c + e\right] \end{aligned} \quad (13)$$

where $\varepsilon = \frac{\eta_w}{\eta_p} > 0$.

Based on the above theoretical modeling, next we analyze the learning dynamics of SA-IGA qualitatively as follows.

**Theorem 1** *SA-IGA has non-linear dynamics when $u_r \neq u_c$.*

***Proof 1*** *From the system of differential equations in (13), it is straightforward to verify that the dynamics of SA-IGA learners are non-linear when $u_r \neq u_c$ due to the existence of $w_r p_c$, $w_c p_r$ or $p_r p_c$ in all equations.*

Since SA-IGA's dynamics are non-linear when $u_r \neq u_c$, in general we cannot obtain a closed-form solution, but we can still resort to solve the equations numerically to obtain useful insight of the system's dynamics. Moreover, a wide range of of important games fall into the category of $u_r = u_c$, in which the system of equations become linear. Therefore, it allows us to use dynamic system theory to systematically analyze the underlying dynamics of SA-IGA.

**Theorem 2** *SA-IGA has linear dynamics when the game itself is symmetric.*

| 1's payoff<br>2's payoff | | Agent 2's actions | |
|---|---|---|---|
| | | action 1 | action 2 |
| Agent 1's actions | action 1 | a/a | c/d |
| | action 2 | d/c | b/b |

**Table 2***:* The General Form of a Symmetric Game

***Proof 2*** *A two-player two-action symmetric game can be represented in Table 2 in general. It is obvious to check that it satisfies the constraint of $u_r = u_c$, given that $u_i = r_{11}^i + r_{22}^i - r_{12}^i - r_{21}^i$, $i \in \{r, c\}$. Thus the theorem holds.*

## 3.2 Dynamics Analysis of SA-IGA

Previous section mainly analyzed the dynamics of SA-IGA in a qualitative manner. In this section, we move to provide detailed analysis of SA-IGA's learning dynamics in two representative games: the Prisoner's Dilemma game (Table 3) (as a symmetric game example ) and Coordination game (Table 4) (as an asymmetric game example). Specifically we analyze the SA-IGA's learning dynamics by identifying the existing equilibrium points, which provides useful insights into understanding of SA-IGA's dynamics.

**Theorem 3** *The dynamics of SA-IGA algorithm under Prisoner's Dilemma (PD) game have three types of equilibrium points:*

1. $(0, 0, w_r^*, w_c^*)$*, where* $w_r^*, w_c^* < min\left\{\frac{2(T-R)}{T-S}, \frac{2(P-S)}{T-S}\right\}$;
2. $(1, 1, w_r^*, w_c^*)$*, where* $w_r^*, w_c^* > max\left\{\frac{2(T-R)}{T-S}, \frac{2(P-S)}{T-S}\right\}$;
3. $(p^*, p^*, w^*, w^*)$*, others*

*The first and second types of equilibrium points are stable, while the last is not. We say an equilibrium point is stable if once the strategy starts "close enough" to the equilibrium (within a distance $\delta$ from it), it will remain "close enough" to the equilibrium point forever.*

| 1's payoff<br>2's payoff | | Agent 2's actions | |
|---|---|---|---|
| | | C | D |
| Agent 1's actions | C | R/R | S/T |
| | D | T/S | P/P |

**Table 3***:* The Prisoner's Dilemma Game(where $T > R > P > S$)

***Proof 3*** *Following the system of differential equations in Equations (13), we can express the dynamics of SA-IGA in PD game as follows:*

$$\begin{aligned}
\dot{p}_r &= (u)\, p_c + \frac{T-S}{2} w_r + S - P \\
\dot{p}_c &= (u)\, p_r + \frac{T-S}{2} w_c + S - P \\
\dot{w}_r &= \varepsilon\,(S-T)\,(p_r - p_c) \\
\dot{w}_c &= -\varepsilon\,(S-T)\,(p_r - p_c)
\end{aligned} \tag{14}$$

*where $\varepsilon = \frac{\eta_w}{\eta_p} > 0$,$u = R + P - S - T$.*

*We start with proving the last type of equilibrium points: If there exit an equilibrium $eq = (p_r^*, p_c^*, w_r^*, w_c^*)^T \in (0, 1)^4$, then we have $\dot{p}_i(eq) = 0$ and $\dot{w}_i(eq) = 0$, $i \in \{r, c\}$. By solving the above equations, we have $p_r^* = p_c^* = \frac{S-T}{2u} w^* + \frac{P-S}{u}$ and $w^* = w_r^* = w_c^*$. Since $p_r^*, p_c^* \in (0, 1)$, then we have,*

$$\begin{aligned}
w_r, w_c &> min\left\{\frac{2(T-R)}{T-S}, \frac{2(P-S)}{T-S}\right\} \\
w_r, w_c &< max\left\{\frac{2(T-R)}{T-S}, \frac{2(P-S)}{T-S}\right\}
\end{aligned}$$

*Then $eq = (p_r^*, p_c^*, w_r^*, w_c^*)^T$ is an equilibrium. The stability of $eq$ can be verified using theories of non-linear dynamics[16]. By expressing the unconstrained update differential equations in the form of $\dot{x} = Ax + B$, we have*

$$A = \begin{bmatrix} 0 & u & T-S & 0 \\ u & 0 & 0 & T-S \\ \varepsilon(S-T) & \varepsilon(T-S) & 0 & 0 \\ \varepsilon(T-S) & \varepsilon(S-T) & 0 & 0 \end{bmatrix}$$

*After calculating matrix $A$'s eigenvalue, then we have $\lambda_1 = 0$, $\lambda_2 = u$, $\lambda_3 = -\frac{u}{2} + k$ and $\lambda_4 = -\frac{u}{2} - k$, where $k$ is a constant. Since there exist an eigenvalue $\lambda > 0$, the equilibrium $eq$ is not stable.*

*Next we turn to prove the first type of equilibrium. In this case, we need to put the projection function back since we are dealing with boundary cases. If $p_i = 0, i \in \{r, c\}$, according to the known conditions, we have $w_r, w_c < min\left\{\frac{2(T-R)}{T-S}, \frac{2(P-S)}{T-S}\right\}$. Combined with the unconstrained update differential equations, we have $\lim_{p_i} \dot{p}_i < 0$, then $p_i$ remains unchanged. And because $p_r = p_c = 0$, then for $\forall w_i \in [0, 1]$, $\dot{w}_i((0, 0, w_r^*, w_c^*)) = 0$, then $((0, 0, w_r^*, w_c^*))$ is an equilibrium.*

*Because $w_r, w_c < min\left\{\frac{2(T-R)}{T-S}, \frac{2(P-S)}{T-S}\right\}$, there exist a $\delta > 0$, and a set $U(eq, \delta) = \left\{x \in [0, 1]^4 \,|\, |x - eq| < \delta\right\}$, that for $\forall x \in U(eq, \delta)$, $\lim_{p_i} \dot{p}_i < 0$. Thus p will stablize on the point of 0. Also, because*

$$\lim_{t\to 0} \dot{w}_i = (S-T) \lim_{t\to 0} (p_r - p_c) = (S-T) \lim_{t\to 0} (0 - 0) = 0$$

*then $w$ is also stable, and thus the equilibrium $eq$ is stable.*

*The second type of equilibrium can be proved similarly, which is omitted here.*

Intuitively, for a PD game, from Theorem 3, we know that if both SA-IGA players are initially sufficiently social-friendly (the value of w is large than a certain threshold), then they will always converge to mutual cooperation of $(C, C)$. In other words, given that the value of $w$ exceeds certain threshold, the strategy point of $(1, 1)$ (or $(C, C)$) in the strategy space is asymptotically stable. If both players start with a low socially-aware degree ($w$ is smaller than certain threshold), then they will always converge to mutual defection of $(D, D)$ eventually. For the rest of cases, there exist infinite number of equilibrium points in-between the above two extreme cases, all of which are not stable.

Next we turn to analyze the dynamics of SA-IGA playing coordination game by identifying all equilibrium points. The general form of a coordination game is shown in Table 4. Intuitively, both Nash equilibria (C, C) and (D, D) can be part of the equilibrium points depending on the agents' social-aware degrees. Formally we have,

**Theorem 4** *The dynamics of SA-IGA algorithm under a coordination game have three types of equilibrium points:*

1. $(0, 0, w_r^*, w_c^*)$*, with* $w_r^* = 1 \wedge w_c^* = 0$ *when* $P > p > s$*;* $w_r^* = 0 \wedge w_c^* = 1$ *when* $T < P < p$*; and* $\left(\frac{s-S}{2} w_r^* < P - S\right) \wedge \left(\frac{T-t}{2} w_c^* < p - t\right)$ *when* $P = p$*;*

| 1's payoff<br>2's payoff | | Agent 2's actions | |
|---|---|---|---|
| | | C | D |
| Agent 1's actions | C | R/r | S/s |
| | D | T/t | P/p |

**Table 4***:* The General Form of a Coordination Game (where $R > T \wedge P > S$ and $r > s \wedge p > t$)

2. $(1, 1, w_r^*, w_c^*)$*, with* $w_r^* = 1 \wedge w_c^* = 0$ *when* $R > r > t$*;* $w_r^* = 0 \wedge w_c^* = 1$ *when* $T < R < r$*; and* $\left(\frac{T-t}{2} w_r^* < R - T\right) \wedge \left(\frac{S-s}{2} w_c^* < r - s\right)$ *when* $R = r$*;*
3. *others non-boundary equilibrium points* $(p_r^*, p_c^*, w_r^*, w_c^*)$

*The first and second types of equilibrium points are stable, while the last non-boundary equilibrium points are not. The definition of a stable equilibrium point is the same as Theorem 3.*

***Proof 4*** *Following the system of differential equations in Equations (13), we can express the dynamics of SA-IGA in coordination game as follows:*

$$\begin{aligned} \dot{p}_r &= \left(u_r + \frac{u_c - u_r}{2} w_r\right) p_c + \frac{c_c - c_r}{2} w_r + c_r \\ \dot{p}_c &= \left(u_c + \frac{u_r - u_c}{2} w_c\right) p_r + \frac{d_r - d_c}{2} w_c + d_c \\ \dot{w}_r &= \varepsilon \left[\left(u_r - u_c\right) p_r p_c + \left(c_r - c_c\right) p_r + \left(d_c - d_r\right) p_c + e\right] \\ \dot{w}_c &= -\dot{w}_r \end{aligned} \tag{15}$$

*where* $\varepsilon = \frac{\eta_w}{\eta_p} > 0$*,*$u_r = R+P-S-T > 0$*,* $u_c = r+p-s-t > 0$*,* $c_r = S - P$*,* $c_c = s - p$*,* $d_r = T - P$*,* $d_c = t - p$*, and* $e = P - p$*. We can see that the dynamic of coordination game is nonlinear when* $u_r \neq u_c$*. We start with proving the last type of equilibrium points first:*

*If there exit a equilibrium* $eq = (p_r^*, p_c^*, w_r^*, w_c^*)^T \in (0, 1)^4$*, then there have* $\dot{p}_i(eq) = 0$ *and* $\dot{w}_i(eq) = 0$*,* $i \in \{r, c\}$*. By linearizing the unconstrained update differential equations into the form of* $\dot{x} = Ax + B$ *in point* $eq = (p_r^*, p_c^*, w_r^*, w_c^*)^T$*, we have*

$$A = \begin{bmatrix} 0 & u_r^* & a_{13} & 0 \\ u_c^* & 0 & 0 & a_{24} \\ -\varepsilon a_{13} & \varepsilon a_{24} & 0 & 0 \\ \varepsilon a_{13} & -\varepsilon a_{24} & 0 & 0 \end{bmatrix}$$

*where* $u_r^* = u_r + \frac{u_c - u_r}{2} w_r^*$*,* $u_c^* = u_c + \frac{u_r - u_c}{2} w_c^*$*,* $c_r^* = \frac{c_c - c_r}{2} w_r^* + c_r$*, and* $d_c^* = \frac{d_r - d_c}{2} w_c^* + d_c$*. The parameters* $a_{ij}$ *are represented as functions of* $p_r^*, p_c^*, w_r^*$ *and* $w_c^*$*. Without loss of generality, we set* $u_r \geq u_c$*. Because of* $u_r \geq u_c > 0$*, and* $w_r^*, w_c^* \in [0, 1]$*, we have* $u_r^* \in [\frac{u_c + u_r}{2}, u_r]$ *and* $u_c^* \in [u_c, \frac{u_c + u_r}{2}]$*, which means* $u_c^* > u_c^* > 0$*.*

*After calculating matrix A's eigenvalue in Matlab, we have an eigenvalue* $\lambda_1 = 0$*, a eigenvalue* $\lambda_2$ *with its real part* $Re(\lambda_2) > 0$*, an eigenvalue* $\lambda_3$ *with* $Re(\lambda_3) < 0$ *and an eigenvalue* $\lambda_4$ *close to* $0$*. Since there exists an eigenvalue* $\lambda > 0$*, the equilibrium* $eq$ *is not stable[16].*

*Next we turn to prove the first type of equilibrium. In this case, we need to put the projection function back since we are dealing with boundary cases.*

*For the case* $P > p > s$*, we have* $V_i^{idv}(eq) > V_i^{soc}(eq)$*, thus* $\dot{w}_r(eq) > 0$ *and* $\dot{w}_c(eq) < 0$*, which means* $w_r$ *and* $w_c$ *will keeps* $w_r = 1$ *and* $w_c = 0$*. Because* $\dot{p}_r(eq) = \frac{s - p + S - P}{2} < 0$ *and* $\dot{p}_c(eq) = t - p < 0$*, then* $p_r$ *and* $p_c$ *will keeps* $p_r = 0$ *and* $p_c = 0$*. According to the continuity theorem of differential equations [7],* $(0, 0, 1, 0)$ *is a stable equilibrium. The case* $p > P > T$ *can be proved similarly, which is omitted here.*

*For the case* $P = p$*, we have* $V_i^{idv} = V_i^{soc}$*, then* $\dot{w}_r(eq) = -\dot{w}_c(eq) = \varepsilon\left(V_r^{idv} - V_r^{soc}\right) = 0$*. Because* $\left(\frac{T-t}{2} w_c^* < p - t\right)$*, we have* $\dot{p}_r = \frac{T-t}{2} w_c^* + t - p < 0$*. Because* $\left(\frac{s-S}{2} w_r^* < P - S\right)$*, we have* $\dot{p}_c = \frac{s-S}{2} w_c^* + S - P < 0$*. According to the continuity theorem of differential equations,* $(0, 0, w_r^*, w_c^*)$ *is a stable equilibrium. The stability of the second type of equilibrium points can be proved similarly, which is omitted here.*

## 4 A Practical Algorithm

In SA-IGA, each agent needs to know the policy of its opponent and the payoff matrix, which are usually not available before a repeated game starts. Based on the idea of SA-IGA, we relax the above assumptions and propose a practical multiagent learning algorithm called Socially-Aware Policy Gradient Ascent (SA-PGA). The overall flow of SA-PGA is shown in Algorithm 1. In SA-PGA, each agent only needs to observe the payoffs of both agents by the end of each round.

**Algorithm 1** SA-PGA for player $i$

1: Let $\alpha \in (0, 1)$ and $\delta_p, \delta_w \in (0, 1)$ be learning rates.
2: Initialize $Q_i^{idv}(a) \leftarrow 0$, $Q_i^{op}(a) \leftarrow 0$,$Q_i(a) \leftarrow 0$, $w_i \leftarrow 0.5$, $\pi_i(a) \leftarrow \frac{1}{|A_i|}$.
3: **repeat**
4: Select action $a \in A_i$ according to mixed strategy $\pi_i$ with suitable exploration.
5: Observing reward $r$ and its opponent's reward $r'$, $Q_i^{idv}(a) \leftarrow (1 - \alpha) Q_i^{idv}(a) + \alpha r$, $Q_i^{op}(a) \leftarrow (1 - \alpha) Q_i^{op}(a) + \alpha r'$,
6: $Q_i(a) \leftarrow \left(1 - \frac{w}{2}\right) Q_i^{idv}(a) + \frac{w}{2} Q_i^{op}(a)$,
7: Average payoff $V_i = \sum_{a \in A_i} \pi_i(a) Q_i(a)$
8: **for** each action $a \in A_i$ **do**
9: $\pi_i(a) \leftarrow \pi_i(a) + \delta_p \left(Q_i(a) - V_i(s)\right)$
10: **end for**
11: $\pi_i \leftarrow \Pi_{\triangle}[\pi_i]$
12: $V_i^{idv} = \sum_{a \in A_i} \pi_i(a) Q_i^{idv}(a)$
13: $V_i^{op} = \sum_{a \in A_i} \pi_i(a) Q_i^{op}(a)$
14: $V_i^{soc} = \frac{1}{2}\left(V_i^{idv} + V_i^{op}\right)$
15: $w_i \leftarrow w_i + \delta_w \left(V_i^{idv} - V_i^{soc}\right)$
16: **until** the repeated game ends

In SA-IGA, we know that agent $i$'s policy (the probability of selection each action) is updated based on the partial derivative of the expected value $V_i$, while the social attitude $w$ is adjusted according to the relative value of $V_i^{idv}$ and $V_i^{soc}$. Here in SA-PGA, we first estimate the value of $V_i^{idv}$ and $V_i^{op}$ using Q-values, which are updated based on the immediate payoffs received during repeated interactions. Specifically, each agent $i$ keeps a record of the Q-value of each action for both its own and its opponent ($Q_i^{idv}$ and $Q_i^{op}$) (Line 2). Both Q-values are updated following Q-learning update rules accordingly by the end of each round (Line 5). The overall Q-value of each agent is calculated as the weighted average of $Q_i^{idv}$ and $Q_i^{op}$ weighted by its social attitude $w$ (Line 6). Based on the Q-values, we estimate the value of $V_i$ in SA-IGA as the expected Q-value over all actions given the current policy (Line 7). However, $V_i$ is simply an estimated value instead of a function which cannot be differentiated.

To obtain the derivative of $V_i$ with respect to different actions, we estimate it as the difference between each action's Q-value and the expected Q-value over all actions (the value of $V_i$) (Line 9). Agent $i$'s probability of selecting an action is updated in the direction of the estimated derivative of the action's expected value (Line 8-10). After that, agent $i$'s policy is mapped back to the valid probability space (Line 11). Similarly, the expected individual payoff and its opponent's payoff when agent $i$ plays policy $\pi_i$ are estimated based on its current policy and Q-values (Line 12-13). The value of $V_i^{soc}$ is calculated as the average between $V_i^{idv}$ and $V_i^{op}$ (Line 14). Finally, the social attitude of agent $i$ is updated in the same way as we introduced in SA-IGA based on the estimated $V$-values (Line 15). The updating direction of $w_i$ is estimated as the difference between $V_i^{idv}$ and $V_i^{soc}$.

## 5 Experimental Evaluation

We start the performance evaluation with analyzing the learning performance of SA-PGA under two-player two-action repeated games.

In general a two-player two-action game can be classified into three categories[18]:

**Category 1:** $(r_{11}^r - r_{21}^r)(r_{12}^r - r_{22}^r) > 0$ or $(r_{11}^c - r_{12}^c)(r_{21}^c - r_{22}^c) > 0$. In this case, each player has a dominant strategy and thus the game only has one pure strategy NE.

**Category 2:** $(r_{11}^r - r_{21}^r)(r_{12}^r - r_{22}^r) < 0$ and $(r_{11}^c - r_{12}^c)(r_{21}^c - r_{22}^c) < 0$ and $(r_{11}^r - r_{21}^r)(r_{12}^c - r_{22}^c) > 0$. In this case, there are two pure strategy NEs and one mixed strategy NE.

**Category 3:** $(r_{11}^r - r_{21}^r)(r_{12}^r - r_{22}^r) < 0$ and $(r_{11}^c - r_{12}^c)(r_{21}^c - r_{22}^c) < 0$ and $(r_{11}^r - r_{21}^r)(r_{12}^c - r_{22}^c) < 0$. In this case, there only exists one one mixed strategy NE.

where $r_{ij}^r$ and $r_{ij}^r$ are payoffs of player $r$ and player $c$ respectively when player $r$ takes action $i$ while player $c$ takes action $j$. We select one representative game for each category for illustration.

### 5.1 Category 1

For category 1, we consider the PD game as shown in Table 1. In this game, both players have one dominant strategy $D$, and $(D, D)$ is the only pure strategy NE, while there also exists one socially optimal outcome $(C, C)$ under which both players can obtain higher payoffs.

Figure 1(a) show the learning dynamics of the practical SA-PGA algorithm playing the PD game. The x-axis $p1$ represents player 1's probability of playing action $C$ and the y-axis $p2$ represents player 2's probability of playing action $C$. We randomly selected 20 initial policy points as the starting point for the SA-PGA agents. We can observe that the SA-PGA agents are able to converge to the mutual cooperation equilibrium point starting from different initial policies.

Figure 1(b) illustrates the learning dynamics predicted by the theoretical SA-IGA approach. Similar to the setting in Figure 1(a), the same set of initial policy points are selected and we plot all the learning curves accordingly. We can see that for each starting policy point, the learning dynamics predicted from the theoretical SA-IGA is well consistent with the learning curves from simulation. This indicates that we can better understand and predict the dynamics of SA-PGA algorithm using its corresponding theoretical SA-IGA model.

### 5.2 Category 2

For category 2, we consider the CG game as shown in Table 5. In this game, there exist two pure strategy Nash equilibria (C, D) and (D, C), and both of them are also socially optimal.

Figure 2(a) illustrates the learning dynamics of the practical SA-PGA algorithm playing a CG game. The x-axis $p1$ represents player 1's probability of playing action $C$ and the y-axis $p2$ represents player 2's probability of playing action $C$. Similar to the case of PD game, 20 initial policy points are randomly selected as the starting points. We can see that the SA-PGA agents can converge to either of the aforementioned two equilibrium points depending on the initial policies they start with.

Figure 2(b) shows the learning dynamics predicted by the theoretical SA-IGA approach. Similar to the setting in Figure 2(a), we adopt the same set of 20 initial policy points for comparison purpose. All the learning curves starting from these 20 policy points are drawn accordingly. We can observe that for each starting policy point, the learning dynamics predicted from the theoretical SA-IGA is well consistent with the learning curves obtained from simulation. Therefore, the theoretical model can facilitate better understanding and predicting the dynamics of SA-PGA algorithm.

| 1's payoff<br>2's payoff | | Agent 2's actions | |
|---|---|---|---|
| | | C | D |
| Agent 1's actions | C | 3/4 | 0/0 |
| | D | 0/0 | 4/3 |

**Table 5***:* Coordination game (Category 2)

### 5.3 Category 3

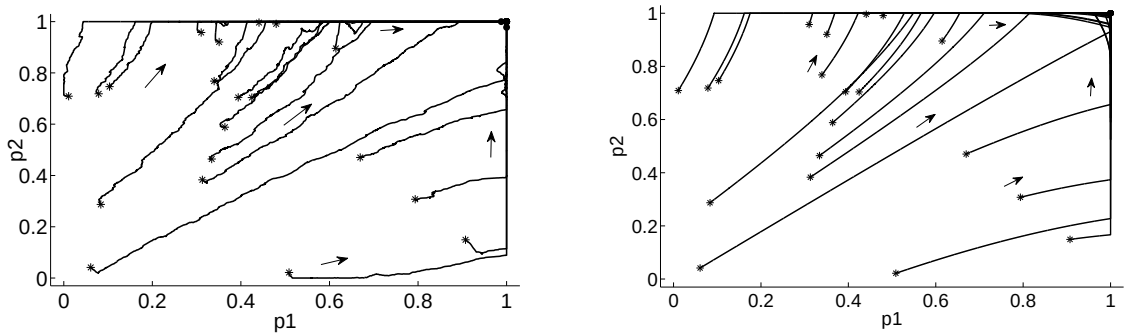
The game we use in Category 3 is shown in Table 6. In this game, there only exist one mixed strategy Nash equilibrium, while the pure strategy outcome $(C, D)$ is socially optimal.

Figure 3(a) illustrates the learning dynamics of the practical SA-PGA algorithm playing the game in Table 6. The x-axis $p1$ and y-axis $p2$ represent player 1's probability of playing action $C$ and player 2's probability of playing action $C$ respectively. Similar to the previous cases, 20 initial policy points are randomly selected as the starting points. From Figure 3(a), we can see that the SA-PGA agents can always converge to the socially optimal outcome $(C, D)$ no matter where the initial policies start with.

Figure 3(b) presents the learning dynamics of agents predicted by the theoretical SA-IGA approach. Similar to the setting in Figure 3(a), we adopt the same set of 20 initial policy points for comparison purpose, and the corresponding learning curves are drawn accordingly. From Figure 3(b), we can observe that for each starting policy point, the theoretical SA-IGA model can well predict the simulation results of SA-PGA algorithm. Therefore, better understanding and insights of the dynamics of SA-PGA algorithm can be obtained through analyzing its corresponding theoretical model.

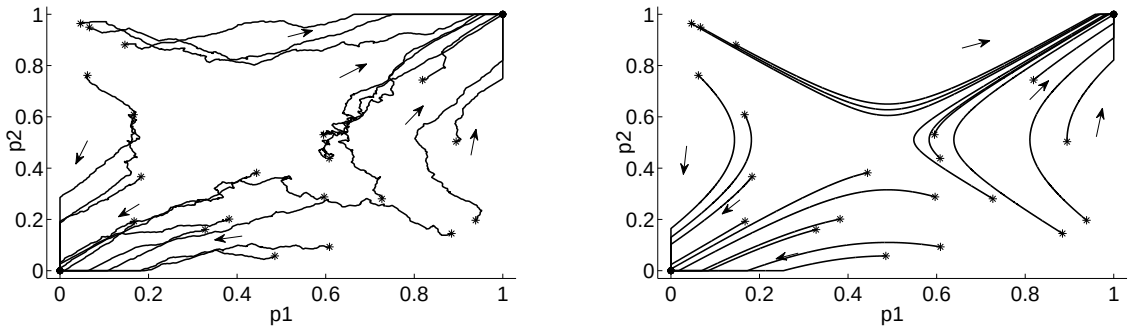
| 1's payoff<br>2's payoff | | Agent 2's actions | |
|---|---|---|---|
| | | C | D |
| Agent 1's actions | C | 3/2 | 4/4 |
| | D | 1/3 | 5/1 |

**Table 6***:* An example game of Category 3



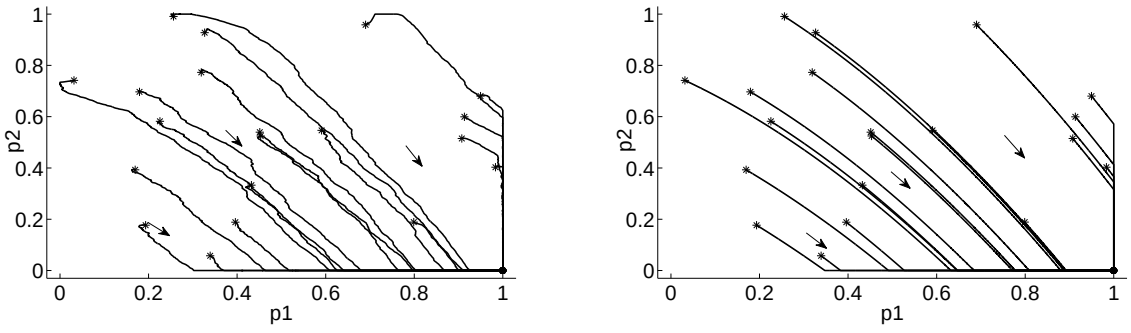

(a) SA-PGA in PD game (b) SA-IGA in PD game

**Figure 1***:* The Learning Dynamics of SA-IGA and SA-PGA in PD game (parameter $w_r(0) = w_c(0) = 0.85$, $\delta_p = 0.001$, $\alpha = 0.8$ and $\varepsilon = 0.02$ )




(a) SA-PGA in CG (b) SA-IGA in CG

**Figure 2***:* The Learning Dynamics of SA-IGA and SA-PGA in coordination game (parameter $w_r(0) = w_c(0) = 0.85$, $\delta_p = 0.001$, $\alpha = 0.8$ and $\varepsilon = 0.02$ )




(a) SA-PGA for the game with one mix NE (b) SA-IGA for the game with one mix NE

**Figure 3***:* The Learning Dynamics of SA-IGA and SA-PGA in game with one mix NE (parameter $w_r(0) = w_c(0) = 0.85$, $\delta_p = 0.001$, $\alpha = 0.8$ and $\varepsilon = 0.02$ )

## 5.4 Performance in General-sum Games

In this section we turn to evaluate the performance of SA-PGA with previous representative learning strategies CJAL [2] and WoLF-PHC [4] in two-player's repeated games under self-play. CJAL is selected since this algorithm is specifically designed to enable agents to achieve mutual cooperation (i.e., maximizing social welfare) instead of inefficient NE for games like prisoner's dilemma. WoLF-PHC is selected as one representative NE-oriented algorithm for baseline comparison purpose. For all previous strategies the same parameter settings used in their original papers are adopted.

We use all possible structurally distinct two-player, two-action conflict games as a testbed for SA-PGA. In each game, each player ranks the four possible outcomes from 1 to 4.We use the rank of an outcome as the payoff to that player for any outcome. We perform the evaluation under 100 randomly generated games with strict ordinal payoffs. We perform 10,000 interactions for each run and the results are averaged over 20 runs for each game.

We compare their performance based on the the following two criteria: utilitarian social welfare and Nash social welfare. Utilitarian social welfare is the sum of the payoffs obtained by the two players in their converged state, averaged over 100 randomly generated games. Nash social welfare is the product of the payoffs obtained by two players in their converged state, averaged over 100 randomly generated games. Both criteria reflect the system-level efficiency of different learning strategies in terms of the total payoffs received for the agents. Besides, Nash social welfare also partially reflects the fairness in terms of how equal the agents' payoffs are. The overall comparison results are summarized in Table 7. We can see that SA-IGA outperforms the previous CJAL strategy under both criteria. The WoLF-PHC strategy is designed to achieve NE and thus can only achieve the same level of performance as adopting NE solutions.

**Table 7***:* Performance comparison with CJAL and WoLF-PHC

| | Utilitarian Social Welfare | Nash Product |
|---|---|---|
| SA-PGA (our strategy) $(w_r(0) = w_c(0) = 0.85)$ | $7.241 \pm 0.003$ | $12.706 \pm 0.015$ |
| CJAL [2] | $6.504 \pm 0.032$ | $10.887 \pm 0.114$ |
| WoLF-IGA [4] | $6.536 \pm 0.004$ | $10.943 \pm 0.145$ |

## 5.5 Against Selfish Agents

If a learning agent is facing selfish agents that attempt to exploit others, one reasonable choice for an effective algorithm is to learn a Nash equilibrium. In this section, we evaluate the ability of SA-PGA against selfish opponents. We adopt the same three representative games used in previous sections as the testbed and the results are given in Figure 4, 5 and 6 respectively. We can observe that for the PD and coordination games, the SA-PGA agent can successfully achieve the corresponding NE solution. This property is desirable since it prevents the SA-PGA agent from being taken advantage by selfish opponents. The results also show how the socially-aware degree $w$ of SA-PGA agent changes, which varies depending on the game structure. For PD and coordination game, a SA-PGA agent eventually behaves as a purely individually rational entity and one pure strategy NE is eventually converged to. In contrast, for the third type of game (Table 6), a SA-PGA agent behaves as a purely socially rational agent and cooperate with the selfish agent towards the socially optimal outcome $(C, D)$ without fully exploiting the opponent. This indicates the cleverness of SA-PGA since higher individual payoff can be achieved under the outcome (C, D) than pursuing Nash equilibrium (C, C).

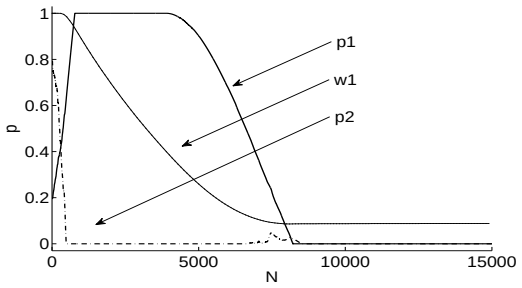



**Figure 4***:* SA-PGA against a selfish agent for in PD game($w_r(0) = 1$, $p_r(0) = 0.2$ and $p_c(0) = 0.8$)

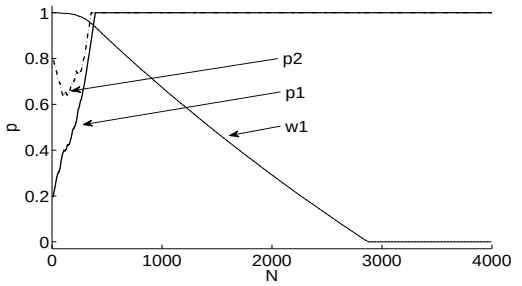



**Figure 5***:* SA-PGA against a selfish agent for in coordination game($w_r(0) = 1$, $p_r(0) = 0.2$ and $p_c(0) = 0.8$)

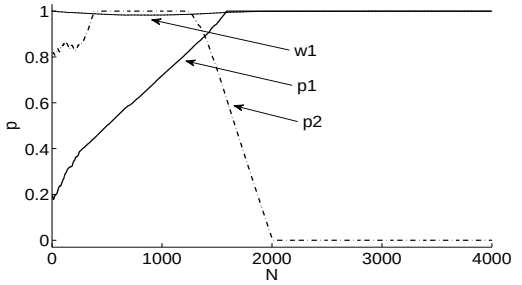



**Figure 6***:* SA-PGA against a selfish agent for the game with only one mix NE($w_r(0) = 1$, $p_r(0) = 0.2$ and $p_c(0) = 0.8$)

## 6 Conclusion and Future Work

In this paper, we proposed a novel way of incorporating social awareness into traditional gradient-ascent algorithm to facilitate reaching mutually beneficial solutions (e.g., (C, C) in PD game). We first present a theoretical gradient-ascent based policy updating approach (SA-IGA) and analyzed its learning dynamics using dynamical system theory. For PD games, we show that mutual cooperation (C,C) is stable equilibrium point as long as both agents are strongly socially-aware. For AC games, either of the Nash equilibria (C,C) and (D,D) can be a stable equilibrium point depending on the agents' socially-aware degrees. Following that, we proposed a practical learning algorithm SA-PGA relaxing the impractical assumptions of SA-IGA. Experimental results show that a SA-PGA agent can achieve higher

social welfare than previous algorithms under self-play and also is robust against individually rational opponents. As future work, more testbed scenarios (e.g., population of agents) will be applied to further evaluate the performance of SA-PGA. Another interesting direction is to investigate how to further improve the convergence rate of SA-PGA.

## REFERENCES


[1] Sherief Abdallah and Victor Lesser, 'A multiagent reinforcement learning algorithm with non-linear dynamics', *Journal of Artificial Intelligence Research*, 521–549, (2008).

[2] D. Banerjee and S. Sen, 'Reaching pareto optimality in prisoner's dilemma using conditional joint action learning', *AAMAS'07*, 91–108, (2007).

[3] Daan Bloembergen, Karl Tuyls, Daniel Hennes, and Michael Kaisers, 'Evolutionary dynamics of multi-agent learning: a survey', *Journal of Artificial Intelligence Research*, 659–697, (2015).

[4] M. H. Bowling and M. M. Veloso, 'Multiagent learning using a variable learning rate', *Artificial Intelligence*, 215–250, (2003).

[5] Lucian Busoniu, Robert Babuska, and Bart De Schutter, 'A comprehensive survey of multiagent reinforcement learning', *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, **38**(2), 156–172, (2008).

[6] Doran Chakraborty and Peter Stone, 'Multiagent learning in the presence of memory-bounded agents', *Autonomous agents and multi-agent systems*, **28**(2), 182–213, (2014).

[7] Levinson N Coddington E A, *Theory of ordinary differential equations*, McGraw-Hill, 1955.

[8] Vincent Conitzer and Tuomas Sandholm, 'Awesome: A general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents', *Machine Learning*, **67**(1-2), 23–43, (2007).

[9] Junling Hu and Michael P Wellman, 'Nash q-learning for general-sum stochastic games', *The Journal of Machine Learning Research*, **4**, 1039–1069, (2003).

[10] M. Lauer and M. Rienmiller, 'An algorithm for distributed reinforcement learning in cooperative multi-agent systems', in *ICML'00*, pp. 535–542, (2000).

[11] M. Littman, 'Markov games as a framework for multi-agent reinforcement learning', in *Proceedings of the 11th international conference on machine learning*, pp. 322–328, (1994).

[12] Michael L Littman, 'Friend-or-foe q-learning in general-sum games', in *ICML*, volume 1, pp. 322–328, (2001).

[13] Laetitia Matignon, Guillaume J Laurent, and Nadine Le Fort-Piat, 'Independent reinforcement learners in cooperative markov games: a survey regarding coordination problems', *The Knowledge Engineering Review*, **27**(01), 1–31, (2012).

[14] Rob Powers and Yoav Shoham, 'Learning against opponents with bounded memory.', in *IJCAI*, volume 5, pp. 817–822, (2005).

[15] Eduardo Rodrigues Gomes and Ryszard Kowalczyk, 'Dynamic analysis of multiagent q-learning with $\varepsilon$-greedy exploration', in *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 369–376. ACM, (2009).

[16] Leonid P Shilnikov, Andrey L Shilnikov, Dmitry V Turaev, and Leon O Chua, *Methods of qualitative theory in nonlinear dynamics*, volume 5, World Scientific, 2001.

[17] Satinder Singh, Michael Kearns, and Yishay Mansour, 'Nash convergence of gradient dynamics in general-sum games', in *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, pp. 541–548, (2000).

[18] Karl Tuyls, Pieter JanT Hoen, and Bram Vanschoenwinkel, 'An evolutionary dynamical analysis of multi-agent learning in iterated games', *Autonomous Agents and Multi-Agent Systems*, **12**(1), 115–153, (2006).

[19] Karl Tuyls, Katja Verbeeck, and Tom Lenaerts, 'A selection-mutation model for q-learning in multi-agent systems', in *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pp. 693–700. ACM, (2003).

[20] C. J. C. H. Watkins and P. D. Dayan, 'Q-learning', *Machine Learning*, 279–292, (1992).

[21] Chongjie Zhang and Victor R Lesser, 'Multi-agent learning with policy prediction', in *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, pp. 927–934, (2010).